An Empirical Study of Feature Extraction Methods for Audio Classification

Charles Parker
Eastman Kodak Company
charles.parker1@kodak.com

Abstract

With the growing popularity of video sharing web sites and the increasing use of consumer-level video capture devices, new algorithms are needed for intelligent searching and indexing of such data. The audio from these video streams is particularly challenging due to its low quality and high variability. Here, we perform a broad empirical study of features used for intelligent audio processing. We perform experiments on a dataset of 200 consumer videos over which we attempt to detect 10 semantic audio concepts.

1 Introduction

In recent years, the increased availability of low-cost video capture devices has resulted in a massive influx of audio and video data. A challenge for the pattern recognition community is to develop intelligent algorithms for indexing and searching this data. As such, it is increasingly important to glean information from the audio stream of consumer-end recording devices, hereafter referred to as "consumer audio".

While there has obviously been a great deal of work around speech recognition and music processing, in this work, we will focus on the more general problem of *semantic audio classification*. That is, we assign semantic labels to recorded audio? Even within this space there has been a fair amount of work in classifying speech vs. non-speech [15, 6], sounds from broadcast audio [5, 2] and music genre identification [12, 8].

Central to nearly all methods of audio classification is the problem of *feature extraction*. That is, given an audio signal, what is the best way to pre-process this signal so that semantic concepts are highly detectable? The current literature provides a variety of methods for feature extraction, but very few empirical comparisons.

It is this task that we will undertake in this paper. Specifically, we engage here in a broad evaluation of "off-the-shelf" feature extraction methods on a comprehensive consumer audio dataset over a large number of audio classes. We believe such an examination is both useful and novel to the literature. We also provide some indication of the difficulty of detecting certain concepts in consumer audio streams.

2 Preliminaries

The problem of *audio frame* classification can be specified as a tuple $\{W, C, Y\}$, where W is a set of k short clips or *frames* of audio, C is a set of n *audio concepts*, and Y is a binary matrix. The matrix Y is size $k \times n$, and the entry $Y_{i,j}$ is 1 if audio concept $\mathbf{c}_j \in C$ is present in audio frame $\mathbf{w}_i \in W$ and 0 otherwise.

The problem, then, is to develop an algorithm that can predict $Y_{i,j}$ given only $\mathbf{w}_i \in W$ for all j. Rather than design n algorithms by hand, we opt to train classifiers [10]. To create a classifier for the concept $\mathbf{c}_j \in C$, we first pick a *feature extraction function*, Φ, and then break W into two sets, W_S and W_T, creating a training set $T = \{(\Phi(\mathbf{w})_i, Y_{i,j})\} : \mathbf{w}_i \in W_T$ and similarly a test set S. We then choose a learning algorithm that will take the training set T as input and output a function $h : \mathbf{w} \mapsto \{0, 1\}$, which is then our predictor.

3 Feature Extraction Methods

A crucial choice, then, is the choice of Φ, the feature extraction function. We hope to choose a method that uses as few features as possible, but preserves all information relevant to semantic understanding.

We will consider the following feature extraction methods, which are representative of the literature: Spectrogram coeffiencts from the short-time Fourier transform, gammatone filterbanks [4], MFCC features [9], mean and variance from wavelet [7] sub-band energies, and a collection of *ad-hoc* features representative of those found in the literature [3]. These methods each produce between 10 and 50 features for classification.

We use the Matlab signal processing toolbox to extract wavelet and spectrogram features. For the Gam-

matone and MFCC features, we use implementations made available by LabROSA[1]. The *ad-hoc* features used are those in [3].

Two other common parameters must be decided: First, for each the processing methods, we must choose a *frame size*. That is, we must choose the length of each short-time "frame" to be analyzed. In addition, we have the availability of Δ-*features* to consider: In certain work [13, 1], the authors choose not only to represent each frame as a set of features, but also to represent the *difference* in feature values from one frame to the next. This will double the size of each feature set.

After constructing a training set, using a feature extraction function as defined above, we must then input this training set to a learning algorithm. We use four learning algorithms for these experiments: Naïve Bayes, logistic regression, adaboosted decision trees, and k-nearest neighbor. We use the Weka [14] implementation for each of these classifiers and use the default options for all of them.

4 Experiment

Our experiment is designed to test the effectiveness of the feature extraction processes outlined above. As such, we use a wide variety of audio concepts for testing and test many possible feature extraction methods.

4.1 Experimental Data

Our experiments is designed to test the effectiveness of these techniques in processing *consumer audio*, and so we gather data from two sources: The first is an in-house camera handout conducted by the Eastman Kodak Company. The second is the popular online video sharing site YouTube. Our dataset consists of 203 videos with an average length of 40 seconds. All videos are consumer-captured, and the audio quality is highly variable.

Each of these videos was hand-labeled for 10 different audio concepts, listed in Table 1 along with the fraction of audio frames in the dataset labeled with that concept. All or any part of a file may take one or more of the labels. Durations as small as 0.1 seconds may take on a label different from the surrounding audio, and the labels may overlap in time.

For each experiment, the data is broken into a training set and a test set (with half of the videos in each set) on a "per-clip" basis. That is, there is never a case where different parts of the same video appear in both sets.

[1] http://labrosa.ee.columbia.edu/

Concept	Frequency	90th %-ile f_1 Ratio
applause	0.024	15.71
baby	0.018	11.78
crowd	0.149	1.78
laughter	0.009	2.72
music	0.131	2.01
parade-drums	0.042	3.21
singing	0.029	3.31
speech	0.150	1.97
water	0.101	3.36
wind	0.024	5.96

Table 1. 90th percentile f_1 ratio and frequency of occurrence for each concept.

4.2 Experimental Procedure

For each experiment, we choose a feature extraction method (described above), a frame size (25, 100 or 250 ms), whether or not to use Δ-features, a learning algorithm, and an audio concept. With these parameter choices decided, we split the videos into training and test sets and sample 5000 frames from the training set, labeling each one with the presence or absence of the chosen audio concept. We train a classifier on this set using the chosen learning algorithm, then sample 5000 frames from the test set for testing. We measure success using the ratio of the classifier f_1 score to the f_1 score of a random classifier.

Because we wish to make general statements about our parameter choices, we perform the experiment 40 times for every possible combination of parameter choices, resulting in 60,000 total experiments.

5 Results and Conclusions

In the results that follow, for each choice of frame size, feature extraction method, Δ-features, and audio concept, we choose the classifier that performed the best of the four we tested. We use this as a proxy for engineering and parameter tuning for each individual concept.

We first present Table 1 showing the 90th percentile f_1 ratio for each of the audio concepts under test. More formally, suppose that, for some parameter choice $\mathbf{p}$, and some concept $\mathbf{c}$, the f_1 ratio achieved by those parameters on that concept is $f(\mathbf{c}, \mathbf{p})$. We report, then, for each $\mathbf{c}$:

$$\min_{\mathbf{p}} f(\mathbf{c}, \mathbf{p}) + 0.9(\max_{\mathbf{p}} f(\mathbf{c}, \mathbf{p}) - \min_{\mathbf{p}} f(\mathbf{c}, \mathbf{p}))$$

Features	25ms	100ms	250ms
MFCC	46.6	58.9	62.1
MFCC$^{\Delta}$	56.7	73.2	70.0
Wavelet	47.5	53.1	55.8
Wavelet$^{\Delta}$	48.5	53.8	59.8
Gammatone	48.7	52.8	56.8
Gammatone$^{\Delta}$	52.7	55.8	57.2
S-gram	49.1	52.2	54.4
S-gram$^{\Delta}$	50.6	55.1	58.6
Ad-hoc	31.6	33.9	35.8
Ad-hoc$^{\Delta}$	36.7	46.6	46.2

Table 2. Average percentile rank over all concepts for various parameter choices. Δ indicates the addtion of Δ-features.

This gives us an idea of the difficulty of detecting certain audio concepts using off-the-shelf methods, assuming one makes good parameter choices.

It is difficult to compare concepts of vastly differing frequencies (due to the relatively much better performance of the random classifier on these concepts), but among the more common concepts (music, speech, and crowd noise), it appears that crowd noise is the most difficult to detect, while speech and music are somewhat easier. Among the less common concepts (all others), some methods do an impressive job classifying applause, baby noises, and, to a lesser extent, wind. Other concepts appear more difficult to classify reliably.

We also summarize the average performance of each combination over all concepts in Table 2. Here we report the average *percentile rank* for each combination over all concepts. That is, if we have n concepts $\mathbf{c}_1 \dots \mathbf{c}_n$, and a set of m possible permutations of parameters $\mathcal{P} = \{\mathbf{p}_1 \dots \mathbf{p}_m\}$, then we report, for each parameter choice $\mathbf{p}_t$:

$$\frac{\sum_i^n \frac{f(\mathbf{c}_i, \mathbf{p}_t)}{max_{\mathbf{p}_m \in \mathcal{P}} f(\mathbf{p}_m, \mathbf{c}_i) - 1}}{n}$$

We do this because the variance of f_1 ratios is so wide that certain concepts (baby, applause) dominate a simple average. This gives us a broader view of the performances of each combination.

As we can see the MFCC-based feature sets do very well in the comparison. It is also seems clear that the *ad-hoc*-based features do not do as well. Further, it appears that Δ-features do indeed help classification, and that the larger frame sizes outperform the smaller.

To evaluate the statistical relevance of these observations, we will perform Wilcoxon signed-rank tests. We have 300 combinations of parameter choices (frame size, feature extraction method, Δ-features, and concepts), each with an associated f_1 ratio. We will split these ratios into groups for our tests.

Test	p-value
Δ vs. no Δ	$p < 0.05^*$
25ms frame vs. 100ms frame	$p < 0.00001^*$
25ms frame vs. 250ms frame	$p < 0.00001^*$
100ms frame vs. 250ms frame	$p > 0.05$
MFCC vs. Gammatone	$p > 0.05$
MFCC vs. Wavelets	$p > 0.05$
MFCC vs. S-gram	$p < 0.05^*$
MFCC vs. *ad-hoc*	$p < 0.00001^*$

Table 3. Results of statistical tests for several parameter choices. A * indicates significance.

The first split is Δ-features vs. no Δ-features, giving us two sets with 150 samples each. We perform a paired test, pairing parameter choices where the presence of Δ-features is the only difference. We do the same with the three frame sizes, performing tests for each choice of two sizes. Finally, we do the same for the six choices of feature extraction method, performing paired tests of the top choice (MFCC features) against all others.

We show the results in Table 3. We see that the use of Δ- features produces an improvement, as does using one of the larger frame sizes over the smaller one. In addition, MFCC features are not significantly better than wavelet features or gammatone filters but are, however, statistically better than the raw spectrogram and the *ad-hoc* features.

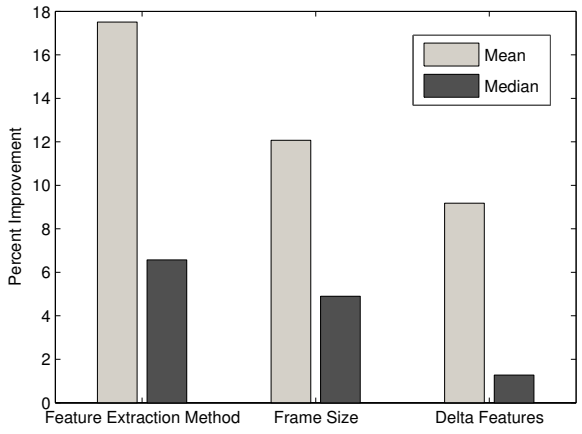

Figure 1. Mean and median improvement due to "correct" parameter choice.

Algorithm	# times best
Naïve Bayes	200
Logistic Regression	7
k-nearest neighbor	90
Adaboost (decision trees)	4

Table 4. Number of times each learning algorithm outperformed all others.

In Figure 1, we show the mean and median improvement in f_1 ratio associated with each "correct" choice of parameter. We define "correct" choices as MFCC, Gammatone, or wavelet feature extraction functions, a 100ms or 250ms frame size, and using Δ features. Incorrect choices are all others. We then compute the mean and median differences for each choice. Choosing the correct feature extraction method is most important, followed by the correct frame size. The use of Δ-features provides less of a performance boost.

We now consider the learning algorithm performances. Table 4 shows the number of tests where each learning algorithm outperformed the other three.

Naïve Bayes and k-nearest neighbor clearly outperform logistic regression and adaboost in general. We note that the former two algorithms regard all dimensions in the feature space equally, while the latter two minimize the influence of less useful dimensions, apparently without success. This may provide some insight as to the usefulness of other classification paradigms on audio signals.

The omission of support vector machines from our list of classifiers is notable, as they are so common in the literature [2, 1]. We found that SVMs in this setting suffered from having unbalanced training data, and parameter tuning had little effect. Though further tuning may have improved them, we were attempting to tune the classifiers as little as possible, and decided to leave SVMs out of these experiments. In addition, we feel that adaboost represents a reasonable substitute [11].

6 Future Work

An obvious direction for future work is to extend the classification from frames to larger chunks of audio. Often, frames are classified in combination with other adjacent frames, requiring us to choose an aggregation method. This can be anything from averaging feature values, to more sophisticated clustering techniques. This work serves as a starting point for that more detailed examination.

References

[1] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo. Large-scale multimodal semantic concept detection for consumer video. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 255–264, 2007.

[2] L. Chen, S. Gunduz, and M. Ozsu. Mixed type audio classification with support vector machine. In *IEEE International Conference on Multimedia and Expo*, pages 781–784, 2006.

[3] T. Giannakopoulos, D. I. Kosmopoulos, A. Aristidou, and S. Theodoridis. Violence content classification using audio features. In *Proceedings of the 4th Helenic Conference on Artificial Intelligence*, pages 502–507, 2006.

[4] B. R. Glasberg and B. C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47(1–2):103–138, 1990.

[5] S. Kiranyaz, A. F. Qureshi, and M. Gabbouj. A generic audio classification and segmentation approach for multimedia indexing and retrieval. In *In Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, EWIMT 2004*, pages 55–62, 2004.

[6] K. Lee and D. Ellis. Voice activity detection in personal audio recordings using autocorrelogram compensation. In *Interspeech ICSLP-06*, pages 1970–1973, 2006.

[7] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[8] M. Mckinney and J. Breebaart. Features for audio and music classification. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 151–158, 2003.

[9] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. In *Pattern Recognition and Artificial Intelligence*, pages 374–388, 1976.

[10] S. Russell and P. Norvig. *Artificial Intelligence, A Modern Approach*, chapter 20, pages 712–762. Pearson Education, second edition, 2003.

[11] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26(5):1651–1686, 1998.

[12] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[13] G. Tzanetakis, G. Essl, and P. Cook. Audio analysis using the discrete wavelet transform. In *Proceedings of the Conference in Acoustics and Music Theory Applications*, 2001.

[14] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann series in data management systems. Morgan Kaufmann, second edition, 2005.

[15] Z. Xiong and T. S. Huang. Boosting speech/non-speech classification using averaged mel-frequency cepstrum coefficients features. In *IEEE Pacific Rim Conference on Multimedia*, pages 573–580, 2002.