# Longer Features: They do a speech detector good

*TJ Tsai*[1,2], *Nelson Morgan*[1,2]

[1]EECS Department, University of California at Berkeley, Berkeley, CA, USA
[2]International Computer Science Institute, Berkeley, CA, USA

tjtsai@eecs.berkeley.edu, morgan@icsi.berkeley.edu

## Abstract

We have incorporated spectrotemporal features in a speech activity detection (SAD) task for the Speech in Noisy Environments 2 (SPINE2) data set. The features were generated by applying 2D Gabor filters to the mel spectrogram in order to measure the strength of various spectral and temporal modulation frequencies in different patches of the spectrogram. Using several different back-ends, the Gabor features significantly outperformed MFCCs, yielding relative reductions in equal error rate (EER) of between 40 and 50%. Compared to the other backends, Adaboost with tree stumps performed particularly well with Gabor features and particularly poorly with MFCCs. An investigation into the reasons for this disparity suggests that the most useful features for SAD incorporate information over longer time scales.

**Index Terms**: spectrotemporal features, speech activity detection

## 1. Introduction

One approach to developing robust features for speech processing tasks is to model higher level representations of audio in the human brain. Neurophysiological experiments have shown that neurons in the primary auditory cortex are tuned to certain types of auditory stimuli (see [1] for related references). The stimulus that causes a high firing rate in a particular neuron is called its spectrotemporal receptive field (STRF), and can be thought of as a particular pattern in a particular patch of the spectrogram. The STRFs exhibit a wide variety of spectrotemporal characteristics and often span much longer periods of time than the time intervals spanned by traditional features like MFCCs. Spectrotemporal features roughly approximate this representation of audio by measuring the strength of various temporal and spectral modulation frequencies in different patches of the spectrogram. Here we explore the use of such features in a noisy speech activity detection task, while also comparing the relative performance of several classification methods.

Earlier studies have also investigated spectrotemporal features for speech/non-speech discrimination. Bach et al. [2] considered the strength of temporal modulations within each spectral channel, and found that the resulting features often had better generalization properties than MFCCs in mismatched train-test conditions. Markaki and Stylianou [3] similarly considered temporal modulations within each spectral channel and employed various methods to reduce the dimensionality of the feature set by minimizing redundancy and maximizing relevance to the target class. The resulting feature set provided classification performance on par with MFCCs, but yielded additional performance gain when combined together with MFCCs. It is useful to point out that filtering temporal modulations within each spectral channel has been shown to improve robustness to noise on other speech processing tasks such as speech recognition [4, 5]. Mesgarani et al. [6] showed that a feature set that considers both temporal and spectral modulation frequencies provides robustness to additive and convolutional noise in a speech discrimination task for 1 second long audio segments. Here we also use features that incorporate both temporal and spectral modulation frequencies, and report performance on a frame-level speech activity detection task for conversations in physically noisy environments (i.e., not adding noise digitally).

## 2. Experimental Setup

This section describes the experimental setup in three parts: the data, the features, and the back-end.

### 2.1. Data

For these experiments we used the second Speech in Noisy Environments (SPINE2) corpus. Related experiments have used TIMIT [2] [6] and RT-03 [3]. The data sets in [2] and [6] are generated by digitally adding noise to clean read speech. This approach has the benefit of being able to see how system performance degrades as the signal-to-noise ratio decreases, as [2] methodically shows for a variety of different noise backgrounds. The disadvantage of this approach is that the artificially generated data is not as accurate a model of speech, as it ignores the Lombard effect (the tendency for people to

speak with more strain and effort in noisy environments), and also misses effects from acoustic reflections in the room. The data set in [3] is a combination of broadcast news and conversational telephone speech, which includes more natural, spontaneous speech but has no explicit noise component. This paper complements previous experiments by using SPINE2, which consists of recorded conversations between two communicators performing a battleship-like task in various noisy environments. While the noise was created artificially (it was played back in the recording rooms over speakers), Lombard effects and some room acoustic phenomena should be captured since the noise was added acoustically. The SPINE2 training and evaluation data sets each contained about 7 hours of audio, and both data sets were equally split among the same 8 noise backgrounds. The backgrounds consisted of military environments ranging from silence to street noise to F16 jet engine noise.

### 2.2. Features

The features explored in this paper are taken from [7]. A general summary of how to compute the features is as follows. (1) Compute the mel spectrogram. (2) Convolve the mel spectrogram with each of the desired 2-dimensional filters. In this case, the set of desired filters consists of 59 Gabor filters (real component only) covering a range of temporal and spectral modulation frequencies. These 59 spectrotemporal filters are shown in figure 1. Note that the size of each filter is such that it includes one and a half cycles of the modulation frequency in both the temporal and spectral dimensions. Intuitively, the resulting Gabor features measure how similar each spectrotemporal filter is to different patches of the spectrogram. The biological analogy of this step is a neuron that fires if it observes a particular pattern in a region of the spectrogram. (3) Perform critical sampling. Since large spectrotemporal filters will yield similar outputs when shifted by only one spectral channel, only 449 of the possible (23 channels * 59 filters =) 1357 features are used at each time index. As a final pre-processing step, mean and variance normalization of the features within each audio file was performed. For a baseline, we also ran experiments using 39 dimensional MFCCs. These baseline features included $\Delta$ and $\Delta$-$\Delta$ components.

### 2.3. Classification Back-Ends

We performed frame-level SAD on the SPINE2 evaluation data set using three different classification algorithms. The first uses a two-state hidden Markov model, where each state is modeled as a mixture of 256 Gaussians. A Viterbi decoding pass is used to determine the best state sequence. The receiver operation characteristic (ROC) curve is generated by sweeping across a range of transition probability values in the acoustic model for

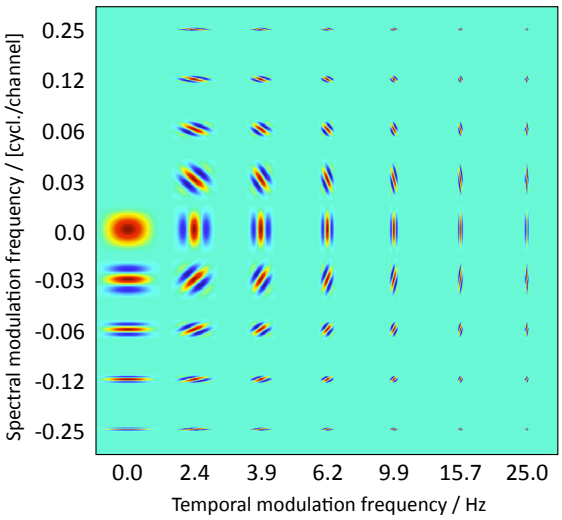


Figure 1: The set of 2D filters applied to the mel spectrogram, shown by temporal and spectral modulation frequencies. Taken with permission from [7].

the speech state. SRI's DECIPHER system was used for this purpose. This system will be referred to as HMM-GMM. The second back-end is a multi-layer perceptron (MLP) with 1 hidden layer and two output nodes. The input layer contains 4 context windows on either side of the current frame (resulting in a total of 9*449=4041 input units for Gabor features, for example), and the size of the hidden layer is selected to ensure that there are approximately 20 training data points per MLP parameter. Given the amount of training data in the SPINE2 corpus, this resulted in hidden layers containing 30 and 345 units for Gabor features and MFCCs, respectively. The MLP output, which approximates the posterior probabilities of speech and nonspeech, is given by applying a softmax nonlinearity to the two output nodes. The speech class probability is then compared to a threshold in order to determine the frame-level hypothesis. This threshold is varied in order to generate the ROC curve. The third back-end is the Adaboost algorithm with tree stumps as weak classifiers. Each tree stump is a single feature compared to a threshold. In other words, the Adaboost classification for frame i is given by

$$f(\vec{x}_i) = sign((\sum_{m=1}^{M} \alpha_m \cdot g_m(\vec{x}_i)) - t) \quad (1)$$

$$g_m(\vec{x}_i) = 2 \cdot \mathbf{1}(\vec{x}_{ij_m} \gtrless \beta_m) - 1 \quad (2)$$

where $\vec{x}_i$ is a vector containing (say) the 449 Gabor features for frame i, M is the number of weak classifiers, $g_m(\vec{x}_i)$ and $\alpha_m$ are the prediction and weight of the $m^{th}$ classifier, and t is the global threshold that sweeps the ROC curve. In equation (2), $j_m$ and $\beta_m$ specify the index of the feature and the threshold for the $m^{th}$

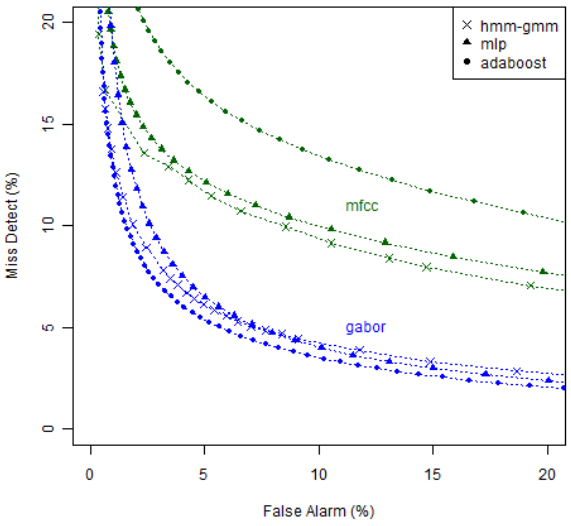


Figure 2: ROC curves for all experiments. The three upper curves (green) correspond to MFCCs, while the three lower curves (blue) correspond to Gabor features.

classifier, respectively. At each iteration in the training phase, $\alpha_m$, $j_m$, $\beta_m$, and the direction of the inequality in (2) is selected to minimize the exponential loss function $L(y, f(\vec{x})) = e^{-y \cdot f(\vec{x})}$ over the training set. M is selected to minimize exponential loss on a separate validation set. For the Adaboost algorithm, small subsets of 20k and 5k randomly selected frames from the SPINE2 training data set were used as training and validation sets, respectively.

## 3. Results

With two different features (MFCCs and Gabor features) and three different back-ends, the results can be summarized by the corresponding six ROC curves, which are shown in figure 2. There are two important aspects of the results to point out.

First, the Gabor features significantly outperform the MFCCs under all back-end configurations. Without even considering the type of back-end, we can see that the Gabor features (three lower curves) outperform the MFCCs (three upper curves) over the range of operating regions between 2 and 15 percent miss detect or false alarm. In fact, the equal error rate (EER) of the worst back-end with Gabor features is still better than the best back-end with MFCCs by an absolute 4.1%. Table 1 shows the EERs of all back-ends for MFCCs and Gabor features, ordered from best to worst.

Second, the Adaboost algorithm is particularly well suited for the Gabor features. Note from table 1 that Adaboost is the best back-end for Gabor features. This is notable because, in contrast, Adaboost is by far the worst back-end for MFCCs. It is also notable because, as mentioned previously, the Adaboost algorithm has a double disadvantage – it has a much smaller training set (20K randomly selected frames, which amounts to $<$1% of the full training set), and it does not use any context frames (as does the MLP). Yet, despite these significant disadvantages, the Adaboost algorithm still outperforms both the HMM-GMM and MLP back-ends for this front end.

| Feature | Back-end | EER | Rel Impr |
|---|---|---|---|
| MFCC | HMM-GMM | 9.9% | |
| MFCC | MLP | 10.2% | -3% |
| MFCC | Adaboost | 12.7% | -28% |
| Gabor | Adaboost | 5.2% | +47% |
| Gabor | HMM-GMM | 5.8% | +41% |
| Gabor | MLP | 5.8% | +41% |

Table 1: EERs for various feature/back-end pairings. The last column indicates relative reduction in EER compared to the MFCC/HMM-GMM baseline.

## 4. Discussion

This section investigates why Adaboost performs so poorly with MFCCs and so well with Gabor features. Three observations help explain this phenomenon.

The first observation is that MFCCs can be considered a special case of spectrotemporal features where the 2D filters are of size nx1, where n is the total number of spectral channels. These "skinny" filters measure spectral modulation frequencies across the entire spectrum within a single time frame. In this regard, MFCCs represent one extreme where the 2D filters are very tall and skinny. This selection means that the resulting features capture information along the entire spectral dimension, but capture no information along the temporal dimension (beyond the current frame). The Gabor filters, on the other hand, have a wide variety of sizes, containing every combination of tall, short, fat, and skinny filters. In particular, this means that many of the filters capture spectral information over more localized regions of the spectrum (rather than over the entire spectrum), and temporal information over broader time intervals. Thus, we can understand the comparison between MFCCs and Gabor features as a comparison between two different sets of spectrotemporal filters, where one set is very constrained and the other set has much more variety.

Secondly, unlike the MLP and HMM-GMM back-ends, the Adaboost algorithm does not incorporate any context information. The MLP backend incorporates context by including the features for the 4 frames before and after the current frame as units in the input layer. The HMM-GMM incorporates context during the Viterbi decoding by estimating the most likely state sequence, rather than just estimating the state of a single isolated

frame. The Adaboost algorithm, however, does not incorporate any context information into its prediction beyond whatever temporal information is contained in the features themselves. So, if the features at a given frame do not contain sufficient temporal information to make a reasonable prediction, we would expect the Adaboost algorithm to perform poorly, as is indeed the case with MFCCs. On the other hand, the Gabor features seem to have captured sufficient temporal context to allow Adaboost to make isolated predictions that are as good as (and in this case, even better than) an estimation of the entire state sequence using the HMM-GMM backend. In this sense, the Gabor features allow the backend to make predictions that are decoupled in time.

Finally, the optimal weak classifiers selected by the Adaboost algorithm (using Gabor features) favored features with low temporal modulation frequencies. One measure of the importance of an input feature to Adaboost is its relative influence [8], computed by considering all the tree stumps that split on that feature and summing the empirical improvement in squared error on the training set as a result of each split. (This measure is the same criterion used to select the feature/threshold pair at each training iteration of Adaboost.) For our Adaboost model, the 173 features capturing temporal modulations of 3.9 Hz and below accounted for more than 95% of the total relative influence of all 449 features. When we trained an Adaboost model on this reduced subset of 173 features, we observed no decrease in system performance (the EER remained at 5.2%). Additionally, the two features with 0 spectral modulation frequency and smallest temporal modulation frequencies (0 Hz and 2.4 Hz) dominated the relative influence, contributing 68% and 11% of the total relative influence, respectively. An Adaboost model trained on only these 2 features yielded an EER of 6.2%. Note that the filters with temporal modulation frequencies of 0, 2.4, and 3.9 Hz span lengths of approximately 1, .7, and .5 seconds, respectively. Because filters with low temporal modulation frequency span longer time intervals, these results strongly suggest that the most useful features for SAD incorporate information over longer time scales, at least for these data.

Putting these 3 observations together, we believe that we can explain why Adaboost performs so poorly with MFCCs and so well with Gabor features. The Gabor feature set represents spectrotemporal filters of many sizes and shapes. Presented with this variety of features, the Adaboost algorithm performs well because it downplays or ignores features that are less useful and gives more weight to more useful features. In this case, it emphasizes features that capture information over long time scales. The MFCCs, on the other hand, exclusively come from tall, skinny spectrotemporal filters – exactly the type of features that Adaboost downplayed with Gabor features. Given the limited range of MFCCs, the Adaboost has poor performance despite its feature selection capability. Because Adaboost does not otherwise incorporate context, the results are especially poor.

## 5. Conclusion

We incorporated Gabor spectro-temporal features (derived from a mel spectrogram) in a noisy speech activity detection task. For each back-end, Gabor features significantly outperformed MFCCs, yielding relative reductions in EER between 40 and 50%. The Gabor features seem particularly well-suited to a simple threshold decision rule, Adaboost with tree stumps, despite using a very small subset of the available training data for this backend only. Results with Adaboost models suggest that the most useful features for SAD use information over long time scales, on the order of .5 to 1 seconds long.

## 6. Acknowledgements

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA or its Contracting Agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. Thanks to Bernd Meyer and Marc Schadler for generously sharing their Gabor feature extraction code, SRI for the use of DECIPHER, Andreas Stolcke for providing SPINE2 training reference labels, and Omid Sadjadi and John Hansen from UT Dallas for providing SPINE2 evaluation reference labels.

## 7. References

[1] Mesgarani, N. and Shamma, S., "Speech Processing with a Cortical Representation of Audio", Proc. of ICASSP, pp. 5872 - 5875, 2011.

[2] Bach, J. H., Anemuller, J. and Kollmeier, B., "Robust speech detection in real acoustic backgrounds with perceptually motivated features", Speech Communication 53, pp. 690-706, 2011.

[3] Markaki, M. and Stylianou, Y., "Discrimination of speech from nonspeech in broadcast news based on modulation frequency features", Speech Communication 53, pp. 726 - 735, 2011.

[4] H. Hermansky and N. Morgan, "RASTA Processing of Speech," IEEE Trans. Speech and Audio Proc., 2(4): 578-589, 1994.

[5] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech," IEEE Proc. ICASSP, Phoenix, Arizona, USA, 1999.

[6] Mesgarani, N., Slaney, M. and Shamma, S. A., "Discrimination of Speech From Nonspeech Based on Multiscale Spectro-Temporal Modulations", in IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, No. 3, May 2006.

[7] Meyer, B. T., Ravuri, S. V., Schadler, M. R., and Morgan, N., "Comparing Different Flavors of Spectro-Temporal Features for ASR", in Proc. of Interspeech, pp. 1269 - 1272, 2011.

[8] J.H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine", Annals of Statistics 29(5): pp. 1189-1232, 2001.