World Academy of Science, Engineering and Technology
International Journal of Physical and Mathematical Sciences
Vol:10, No:5, 2016

Speaker Identification by Atomic Decomposition of Learned Features Using Computational Auditory Scene Analysis Principals in Noisy Environments

Thomas Bryan, Veton Kepuska, Ivica Kostanic

***Abstract*—**Speaker recognition is performed in high Additive White Gaussian Noise (AWGN) environments using principals of Computational Auditory Scene Analysis (CASA). CASA methods often classify sounds from images in the time-frequency (T-F) plane using spectrograms or cochleargrams as the image. In this paper atomic decomposition implemented by matching pursuit performs a transform from time series speech signals to the T-F plane. The atomic decomposition creates a sparsely populated T-F vector in "weight space" where each populated T-F position contains an amplitude weight. The weight space vector along with the atomic dictionary represents a denoised, compressed version of the original signal. The arraignment or of the atomic indices in the T-F vector are used for classification. Unsupervised feature learning implemented by a sparse autoencoder learns a single dictionary of basis features from a collection of envelope samples from all speakers. The approach is demonstrated using pairs of speakers from the TIMIT data set. Pairs of speakers are selected randomly from a single district. Each speak has 10 sentences. Two are used for training and 8 for testing. Atomic index probabilities are created for each training sentence and also for each test sentence. Classification is performed by finding the lowest Euclidean distance between then probabilities from the training sentences and the test sentences. Training is done at a 30dB Signal-to-Noise Ratio (SNR). Testing is performed at SNR's of 0 dB, 5 dB, 10 dB and 30dB. The algorithm has a baseline classification accuracy of ~93% averaged over 10 pairs of speakers from the TIMIT data set. The baseline accuracy is attributable to short sequences of training and test data as well as the overall simplicity of the classification algorithm. The accuracy is not affected by AWGN and produces ~93% accuracy at 0dB SNR.

***Keywords*—**Time-frequency plane, atomic decomposition, envelope sampling, Gabor atoms, matching pursuit, sparse dictionary learning, sparse autoencoder.

I. Introduction

SPEAKER identification in high noise environments is a very challenging machine learning problem in today's information age. Humans have the remarkable ability to follow a conversation in the presence of multiple speakers talking at once. Bergman, a psychologist, described this ability as Audio Scene Analysis (ASA) [1]. CASA is the organization of sounds and application of ASA principles in machine learning algorithms [2]. Bergman proposed humans perceive sounds in time by creating *audio streams*. These audio streams are grouping of sounds that are segmented *sequentially* and *simultaneously*. Sequential groupings connect sense data over time whereas simultaneous grouping connect sounds that "are probably parts of the same sounds" [1].

Zhao et al. implemented a robust CASA algorithm using a bank of 54 gammatone filters to model the human cochlear frequency response [2]. Gammatone Features were extracted from speech by decimating the outputs of each filter to produce a T-F representation or cochleargram. Additionally, Gammatone Frequency Cepstral Coefficients (GFCC) features were derived by direct cosine transform of each decimated filter output. An ideal binary mask was derived by supervised training of a hidden Markov model. The ideal binary mask was used to select GFCC features for classification. The GFCC's were shown to significantly outperform Mel Frequency Cepstral Coefficients for speaker identification using the same binary mask technique for feature selection. Zhao's CASA method processes speech as images in the T-F plane using a cochleargram for the visual representation.

Lee et al. also processed sound as images by applying deep belief neural networks to spectrograms [3]. The neural network was able to learn phones/phonemes from raw spectrograms. The learned features were comparable in classification performance to MFCC's. The learned features were combined with MFCC's to further augment the accuracy of the classifier. The combination of learned features and MFCC's were applied to phoneme classification, speaker identification, music artist and music genera classification with good results.

Gross et al. learned sparse dictionaries directly from audio signals for classification purposes. The approach learns features directly from speech for all possible shifts of the audio signal. The learned basis features were shown to provide better accuracy than MFCC's for speaker identification and music genera classification problems [8].

Olshausen and Fields found that a set of sparse overcomplete basis vectors learned directly from natural images are similar to the neural responses in the ganglion cells of the retina [6]. Moreover, they showed at a linear superposition of a few basis vectors from a large dictionary were able to capture the statistical properties of the images. The basis vectors are very similar to features learned by Independent Component Analysis from natural images [7]. The basis vectors are "edges" that are similar to the rings of 2 dimensional Gabor atoms [6].

Gabor proposed the idea that speech may be represented by

T. Bryan (PhD candidate), V. Kepuska (Associate Professor) and I. Kostnaic (Associate Professor) are with the Electrical and Computer Engineering Department, Florida Institute of Technology, Melbourne, Fl, 32901 USA (phone: 321-626-3912, 321- 674-7183, 321-674-7189; e-mail: jbryan@ my.fit.edu, vkepuska@ fit.edu, kostanic@fit.edu).

a superposition of "quanta of information". He recognized that Fourier methods have a T-F uncertainty in a manner similar to Heisenberg uncertainty in nuclear physics. Gabor introduce the idea of Heisenberg boxes to describe the tiling of energy in the T-F plane [4]. He noted that Fourier methods have a single Heisenberg box size for all frequencies. He proposed the Gabor atom, a sinusoid, modulated by a Gaussian pulse as the "quanta of information". By varying the length of the pulse and the center frequency of each atom, the T-F plane could be tiled with different size Heisenberg boxes.

Mallat and Zhang developed the matching pursuit algorithm to perform atomic decomposition of signals using a redundant dictionary of functions [5]. They showed that Gabor functions, or atoms, can be used to perform an adaptive T-F transform. The algorithm isolates structure in the data that is coherent with Gabor atoms. Gabor's idea of representing speech by quanta's of information is reduced to practice by the matching pursuit algorithm. In a previous paper, a sparse autoencoder was used to learn basis features from 16 Gabor and 16 gammatone "seed atoms" from TIMIT training data [10]. Both the Gabor and gammatone atoms were logarithmically distributed in frequency with increasing center frequencies and bandwidth. The learned basis features were used as "custom atoms" for matching pursuit decomposition. Oracle SNR curves were generated that show the reconstruction SNR verses the original clean speech signals. Fig. 1 shows Oracle reconstructions of the learned basis features verses Gabor and gammatone atoms. The learned basis features significantly outperform the Gabor and gammatone atoms, particularly at lower data compression rates. At -10dB SNR, the Gabor and gammatone have ~8dB of denoising gain whereas, the Gabor and gammatone atoms have ~ 3.5dB of denoising gain. At 0dB SNR, the Gabor and gammatone have ~5dB of denoising gain as compared to the Gabor and gammatone atoms that have ~ 2dB of denoising gain.

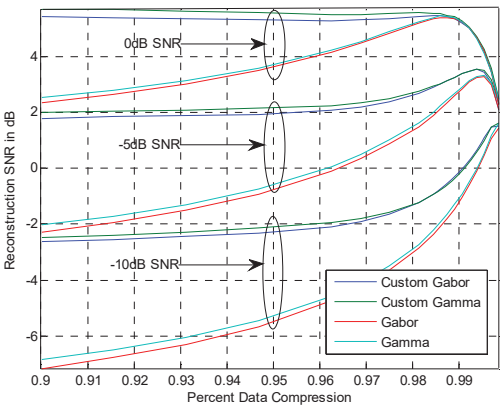

Fig. 1 Basis Feature Denoising Gain verses SNR and data Compression for Basis Features Learned from Gabor and Gammatone Atoms Compared to Gabor and Gammatone Atoms

The SNR performance of basis features learned from Gabor and gammatone atoms are very similar. For simplicity, only the Gabor atoms are used for the remainder of this paper. In another previous paper, a voice activity detector was implemented using 16 Gabor atoms [9]. The VAD detects audio snippets in the presence of AWGN, using matching pursuit decomposition. The detector used the filtered output of the atomic gains in the T-F plane. The detection accuracy shown in Fig. 2 is 70% at 0dB SNR and approximately 95% for SNR's > 8dB.

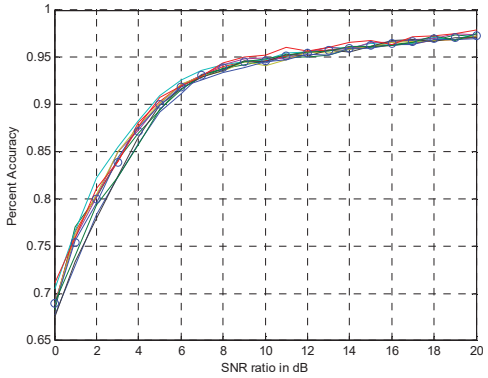

Fig. 2 Audio Segmentation Accuracy, Gabor Atoms, 98.15% Data Compression, 640 Random TIMIT Sentences

The speaker identification algorithm is implemented in the following steps:

For SNR's of 0 dB, 5 dB, 10dB, and 30 dB:

1. Perform Audio Segmentation using matching pursuit with 16 Gabor atoms to generate Audio Snippets. Use, 96.3% data compression to get good denoising gain.
2. Perform a second pass of matching pursuit to generate envelope samples using the Audio Snippets. Generate envelop samples from each sentence in the test and training sets.
3. At 30dB SNR, train the classifier:

a. Learn basis features from envelope samples from the training sentences. Use the best 8 atoms for each sentence. Combine speaker's envelope samples A and B to get 16 envelop samples for unsupervised feature learning.

b. Calculate frequency domain probabilities of T-F atomic indexes.

c. Calculate time domain probabilities of the two atomic indexes with the highest energies. Use time differences along a single atom index in the T-F plane

4. Perform a 3rd pass of matching pursuit using the learned features as custom atoms. Use 90% data compression to get good SNR performance and good statistical samples.
5. For each sentence in the training set, perform classification to calculate training set accuracy.
6. For each sentence in the test set, perform classification to calculate accuracy test set accuracy.

This paper is organized in the following manner. Section II covers the topic of preprocessing the audio signal and segmentation of the audio signal into audio snippets. Section III provides details on matching pursuit envelope sampling. Section IV covers sparse dictionary learning from envelope samples implemented by a sparse autoencoder. Section V describes classification in the T-F plane, and Section VI

discusses the results.

II. Preprocessing and Audio Segmentation

Audio segmentation is a critical part of speaker identification in noisy environments. In a previous paper, a noise tolerant Voice Activity Detector was implemented using matching pursuit atomic decomposition with 16 Gabor or 16 gammatone atoms [9]. The detector filtered a sparsely populated T-F vector of atomic indexes in "weight space". Data denoising and data compression using 16 Gabor and 16 gammatone atoms was shown to achieve peak denoising performance at data compression rates > 95 % for speech [10]. The VAD and denoising/data compression characteristic were very similar for the Gabor and gammatone atoms. Based on the previous research, 16 Gabor atoms are used for the VAD implementation with a data compression of 96.3% in this paper.

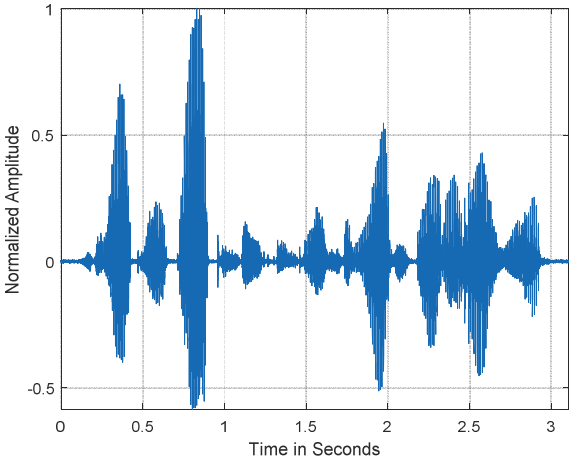

Fig. 3 TIMIT Sentence SA1, Speaker MMDRO, DR1, After Amplitude Normalization and Resampling to 8ksps, 30dB SNR

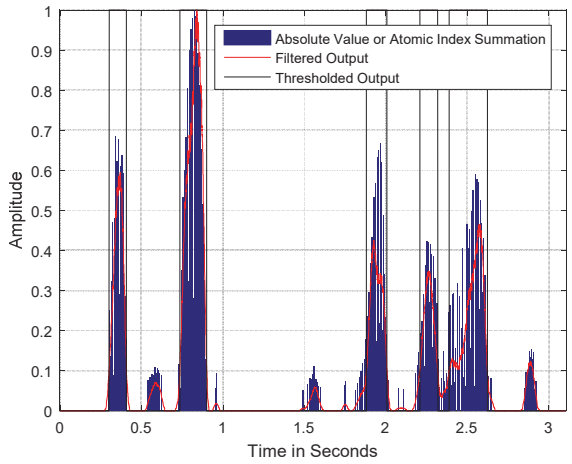

Fig. 4 Audio Segmentation by Matching Pursuit Showing Atomic Indexes in the Time Domain, Filter Output and Thresholded Output at 30dB SNR

The first preprocessing step is to change the sample rate of TIMIT data from 16ksps to 8ksps to minimize simulation time. The TIMIT data is also normalized so that the peak amplitude is 1. A typical TIMIT sentence is shown in Fig. 3. Fig. 4 shows the results of audio segmentation at 30dB SNR. Fig. 5 shows normalized audio snippets.

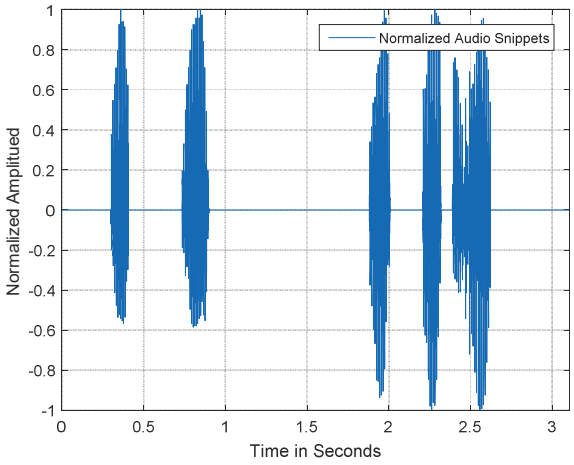

Fig. 5 Normalized Audio Snippets Found by Matching Pursuit Atomic Decomposition using 16 Gabor Atoms at 30dB SNR

Following amplitude normalization and resampling, the speech waveform is segmented into active audio snippets. This process eliminates the silent periods between sounds that would not have any value in classification. The audio segmentation is performed by performing atomic decomposition on the TIMIT data using a set of 16 Gabor atoms. The 16 Gabor atoms are logarithmically distributed in frequency from 300 Hz to 2700 Hz. The sample rate is, F_s= 8ksps. The window length for all Gabor and gammatone atoms is 40.6 mSec which corresponds to 325 samples at 8ksps.

The logarithmically spaced center frequencies ($\boldsymbol{F_c}$) are:

$\boldsymbol{F_c}$ = [299, 347, 402, 465, 538, 624, 722, 836, 968, 1121, 1298, 1502, 1739, 2014, 2332, 2699] Hz.

The Gabor atoms are defined as a sinusoid multiplied by a Gaussian envelope. The same center frequencies $\boldsymbol{F_c}$, for the sinusoid are used for the Gabor atoms. The equation for the Gabor atom is given by,

$$\gamma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/\sigma^2} cos(\omega_c t + \theta) \quad (1)$$

The bandwidth of the Gabor atoms is set by σ, which increases logarithmically with center frequency. The vector $\boldsymbol{\sigma_{cs}}$ was determined empirically to produce high VAD accuracy at 0dB SNR. The Gabor vector $\boldsymbol{\sigma_{cs}}$ is:

$\boldsymbol{\sigma_{cs}}$= [.50, .71, 1.01, 1.44, 2.05, 2.92, 4.16, 5.92, 8.43, 12.01, 17.09, 24.34, 34.65, 49.33, 70.24, 100.00].

The audio segmentation response using 16 Gabor atoms is shown in Fig. 2. The segmentation process generates audio snippets. The minimum size snippet is limited to 50 msec.

This eliminates short spurious bursts that might not be useful for classification. In this example, 5 audio snippets are generated. Out of a total of just over 3.1 seconds of audio data, there are approximately .75 seconds of audio snippets. The final audio preprocessing step is to normalize individual audio snippets to a peak amplitude of 1. This supports uniform amplitude scaling of extracted audio data from each audio snippet. Normalized audio snippets are shown in Fig. 3. Figs. 6-8 show the same set of waveforms for 0 dB SNR. This demonstrates the robustness of the audio segmentation process for low SNR's. Audio segmentation is accurate for the high energy audio snippets, however, for lower energy snippets the detector found 6 audio snippets at 0dB as compared to 5 at 30dB SNR. This behavior affects the accuracy of the classifier as the amount of sampled data is different between high and low SNR samples.

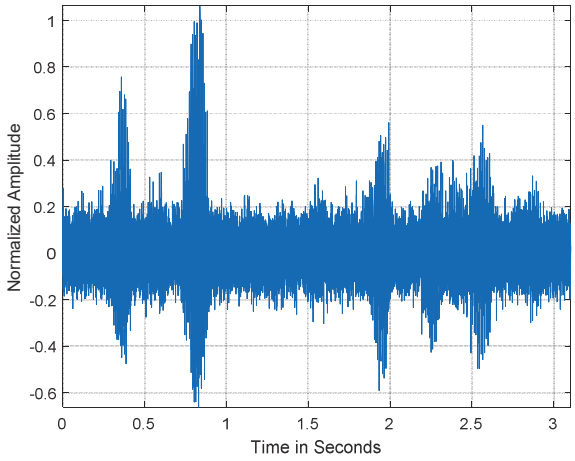

Fig. 6 TIMIT Sentence SA1, Speaker MMDRO, DR1 After Amplitude Normalization and Resampling to 8ksps, 0dB SNR

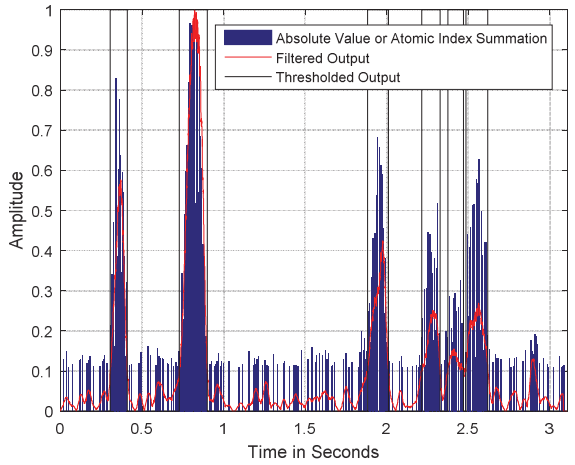

Fig. 7 Audio Segmentation by Matching Pursuit Showing Atomic Indexes in the Time Domain, Filtered Output and Thresholded Output at 0dB SNR

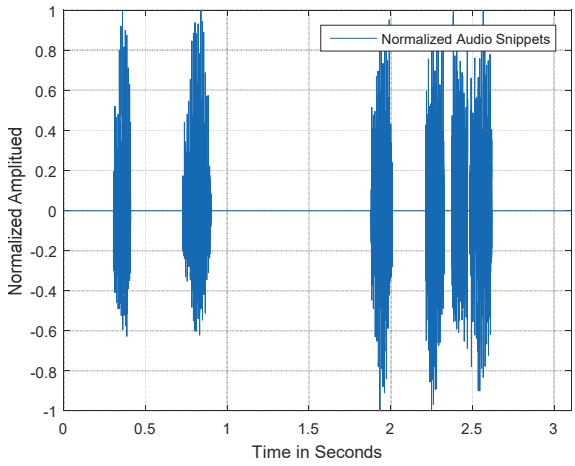

Fig. 8 Normalized Audio Snippets Found by Matching Pursuit Atomic Decomposition using 16 Gabor Atoms at 0dB SNR

III. Matching Pursuit Envelope Sampling of Audio Snippets

Matching pursuit is a greedy algorithm that represents time series data as a linear superposition of fundamental atoms. Matching pursuit follows a simple heuristic of finding correlation peaks between input data and a set of atoms for all possible shifts of each atom. The algorithm finds correlation peaks that represent a minimum mean square error (MMSE) fit between portions of the data and the best atom provided the L_2 norm of the atom is set to 1. The algorithm halts once it reaches stopping goal that is specified by a fixed number of iterations based on the desired data compression.

The audio stream $x(t)$may be represented by a linear superposition of atoms $\varphi_m$from a dictionary $\mathcal{D}$.

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_{m,i} \varphi_m(t - \tau_{m,i}) + e(t) \quad (2)$$

where $e(t)$ is the approximation error or residual, after n iterations of matching pursuit. The input data length is designated by L, so that the number of iterations of the algorithm is given by,

$$n = DL \quad (3)$$

where, D is the data compression.

Simulation parameters:

- The length of the input audio data, L.
- Number of iterations, $n = DL$.
- The number of Gabor and gammatone atoms, $M = 16$.
- The index of the correlation peak,i.
- The amplitude coefficient for the time index i and atom $m$is denoted by $s_{m,i}$.
- The time index i and atom $m$coefficient are denoted by $\tau_{m,i}$.

The steps of matching pursuit envelope sampling are:

1. Initialize the algorithm $\mathcal{R}_0 = x(t)$
2. Compute for all $\varphi_{m,} \in \mathcal{D}$:CORR$(\mathcal{R}_{n-1}, \varphi_{m,}) = |\langle \mathcal{R}_{n-1}, \varphi_{m,} \rangle|$
3. Find the largest inner product, maxArg$\left(\left|\langle \mathcal{R}_{n-1}, \varphi_{m,} \rangle\right|\right)$

4. Extract the audio patch by envelope sampling $\alpha p = x_i(t-\tau_{m,i}) \otimes \varphi_{m\_1}(t-\tau_{m,i})$
5. Where αp = audio patch, and φ_{m_1} is φ_m normalized to 1
6. Compute the new residual, $\mathcal{R}_n = \mathcal{R}_{n-1} - \langle \mathcal{R}_{n-1}, \varphi_{m,} \rangle \varphi_{m,}$
7. Repeat step 2-5 until n iterations of the algorithm are complete.

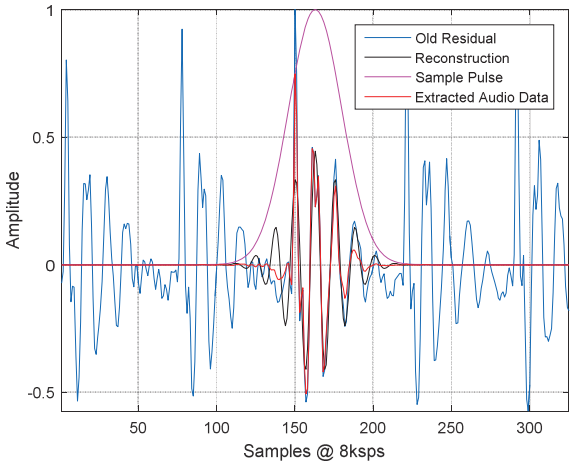

Fig. 9 Envelop Sampling of TIMIT Sentence SA1, Speaker MMDRO, DR1 using Gabor Atom #6 at 30dB SNR

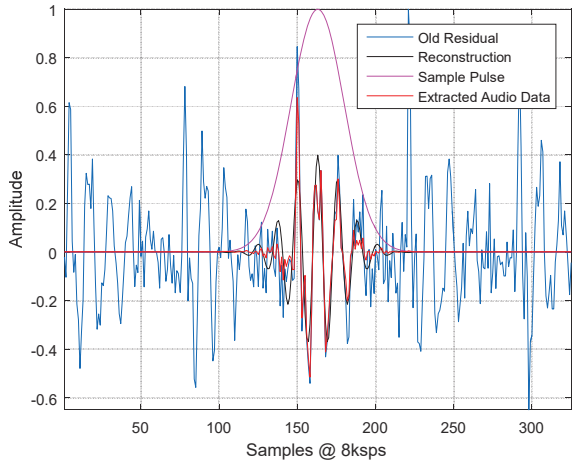

Fig. 10 Envelop Sampling of TIMIT Sentence SA1, Speaker MMDRO, DR1 using Gabor Atom #6 at 0dB SNR

Envelope samples are shown in Figs. 9 and 10 for 30 dB and 0 dB respectively. In these examples, the 6th Gabor atom matches coherent structure in the data. The reconstruction shows that Gabor atom does not match the data exactly. The envelope sample is generated by multiplying the raw data by the Gaussian envelope of the 6th Gabor atom. These examples show the results of the 1st iteration of the algorithm which represent the highest energy portion of the audio signal. Also, the envelope samples are very similar even under noise conditions. Note that the sampler captured a high energy pulse at the beginning of the sample. This will be useful for learning basis vectors from envelope samples data in the next section.

IV. Sparse Dictionary Learning using a Sparse Autoencoder

In order to increase the signal to noise performance of the classifier, a sparse autoencoder is used to learn basis features from the training data. The basis features are used as custom atoms that provide superior denoising and data compression compared to the performance of the Gabor atoms. Envelope samples are extracted from audio snippets by a second application of matching pursuit using 16 Gabor atoms. The envelope samples are arranged in a column vector for processing by a sparse autoencoder. A block diagram of a Sparse Autoencoder (SAE) is shown in Fig. 11. The SAE is a neural network in which the target outputs are equal to the inputs during training. The network represents an input as a superposition of a few basis vectors from a larger set of overcomplete basis vectors. A sparsity parameter punishes overactive nodes in the hidden layer ensuring that the average rate of activation for all nodes is typically <10%. The network has an input layer, a single hidden layer and an output layer. The SAE learns the identity function and performs data compression by 2 mechanisms. This input data is compressed from an input size of 324 to 16 based on using only 16 hidden nodes. Additional data compression results from a sparsity penalty that limits average activation for each hidden node to 1/16. The overall data compression in the SAE is 324:1. The detailed design equations including the cost function and the sparsity equations are found in [10]. The SAE simulation parameters are:

- Number of Inputs = 324
- Number of Hidden Nodes = 16
- Sparsity Parameter $\hat{\rho} = \frac{1}{16} = .0625$
- Sparsity Influence $\beta = 3$;
- Regularization $\lambda = 3.0e\text{-}3$

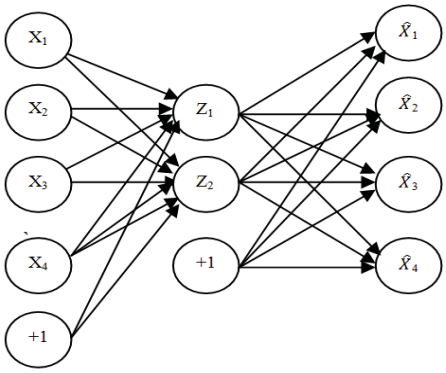

Fig. 11 Sparse Autoencoder Block Diagram

The hidden nodes and the output nodes use a soft limiter based on a sigmoid function that is shown in Fig. 12.

$$f(z) = \text{sigmoid}(z) = \frac{1}{1+e^{-z}} \quad (4)$$

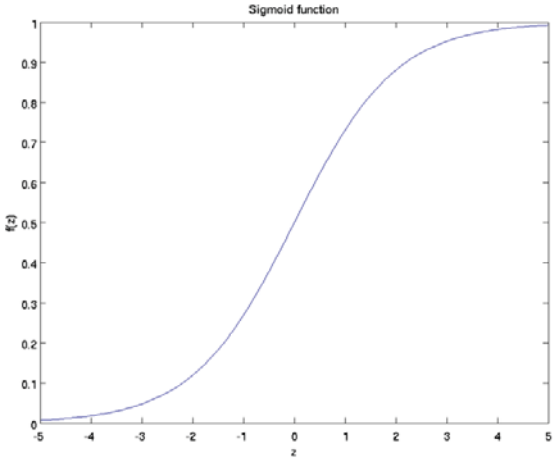

Fig. 12 Sigmoid Soft Limiter Response

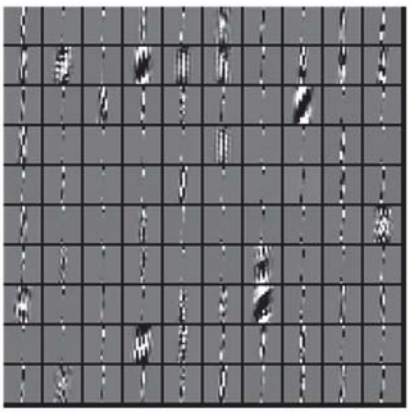

Fig. 13 Random Samples of Speech Viewed as Images by Folding the Time Domain into columns

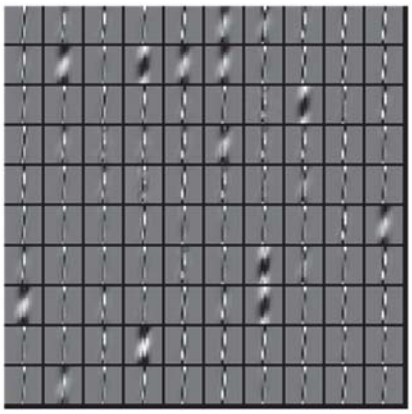

Fig. 14 SAE Reconstruction of the input data from Fig. 13

The sigmoid function is used to facilitate viewing audio data as images, with the darkest portions corresponding to values near 0, while the whitest portions correspond to values near 1. The envelope samples are 324 samples long and are folded into 18 columns of 18 samples per column so that large amounts of audio data can be observed simultaneously. Fig. 13 shows envelope samples that are based on 16 Gabor atoms. Fig. 14 shows the output of the SAE using 16 basis features shown in Fig. 15. The SAE output can be seen to approximate the input data with the image definition being a slightly blurred version of the original input. The SAE is trained by batch backpropagation and the basis features are simply the input weights to the network. The basis features appear as bands when viewed as images. The width of the band is proportional to frequency, while the image intensity is proportional to amplitude.

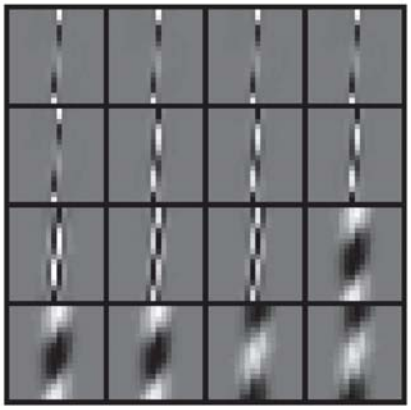

Fig. 15 Basis Features Viewed as Images Learned by a Sparse Autoencoder. Learned from Speakers A and B

V. Classification in the T-F Plane

A final 3rd pass of matching pursuit is applied to the audio snippets using the basis features learned from the training data. The basis vectors are learned at 30dB SNR. This application of matching pursuit uses a data compression rate of 90%. This generates a large number of atomic decomposition components in the T-F plane that produce a rich set of statistics. The atomic indexes are collected from all the audio snippets within a given sentence. Statistical models are generated for each sentence in the training set. Statistics are collected from the test data a SNR's of 0dB, 5dB, 10dB and 30dB using learned basis feature decomposition. The test statistics are compared to the individual sentence training models for speaker identification. The following statistics are generated for classification:

1. Frequency measurements are identified by summing the absolute value across time for each atomic index. Since the atoms are logarithmically distributed in frequency, this will produce an estimate of the total frequency content for each sentence. The amplitude weights represent the energy content for each atom in the T-F plane.
2. Differential time measurements are made by finding the differences between consecutive time indexes for each atom. This is done for the individual snippets. The data is collected for the 2 highest energy atoms in a given snippet. Differential time is used as opposed to absolute time to eliminate large numbers of zero elements in the T-F plane.

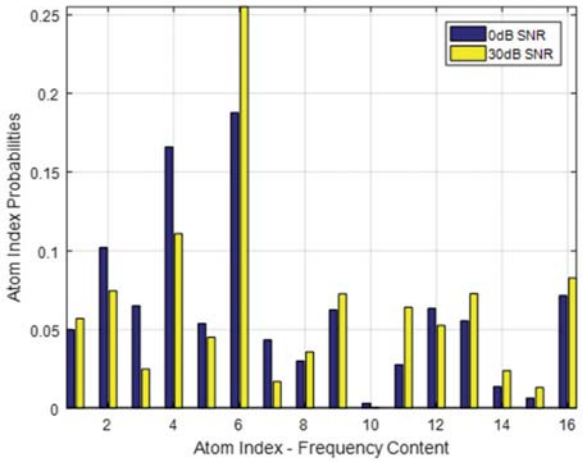

Fig. 16 Example of Atomic Index Probabilities Showing Frequency Content of TIMIT Sentence at 0dB and 30dB SNR's

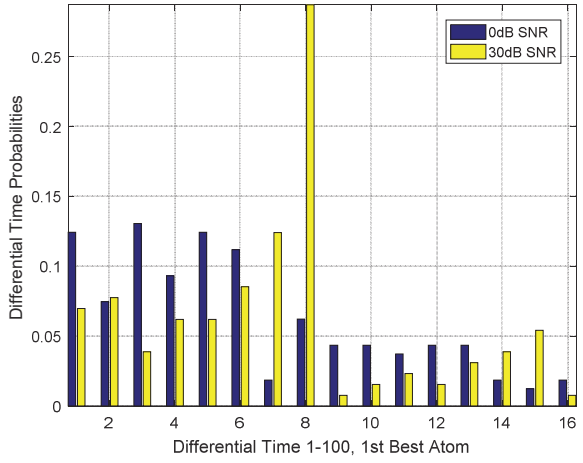

Fig. 17 Example of Differential Time of Highest Energy Atom #6 for a TIMIT Sentence at 0dB and 30dB SNR's

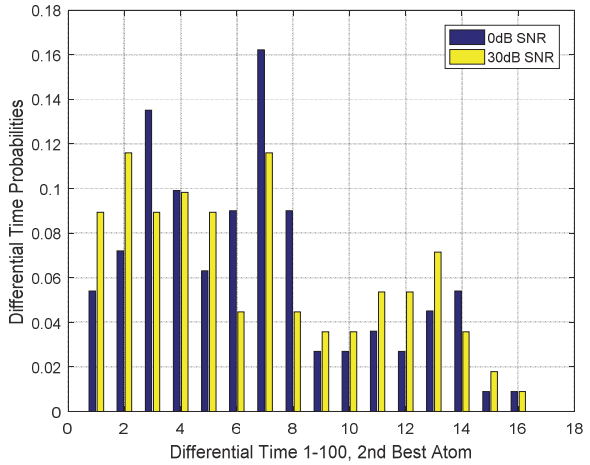

Fig. 18 Example of Differential Time of 2nd Highest Energy Atom #4 for a TIMIT Sentence at 0dB and 30dB SNR's

Examples of statistics for the frequency content for a TIMIT sentence is shown in Fig. 16. The SAE learned features typically have a high energy atom that best matched the speakers voice fundamental frequency response. The differential time distributions for the 1st and 2nd highest energy atoms are shown in Figs. 17 and 18 for atom 6 and 4 respectively. Note that the time differences for the 2nd highest energy atom provide a good match between the training data at 30dB and the test data at 0dB SNR.

The classification was performed on a sentence by sentence basis by finding the minimum Euclidean distance between the measured probabilities and the model probabilities. The classification accuracy averaged over 10 pairs of speakers with SNR's of 30dB, 10 dB, 5dB and 0 dB:

- Atomic Total Energy 81.72%
- Time differences 1st highest energy Atom 84.06%
- Time differences 2nd highest energy Atom 92.81%

VI. Summary and Conclusions

This paper used principals of CASA to perform speaker identification at low SNR's with AWGN. Classification was performed by using statistics of atomic energy and time differences in the T-F plane. The algorithm has identical classification accuracies for SNR's of 0dB, 5dB, 10dB and 30 dB using 2 TIMIT sentences for training and 8 for testing. It is not clear why the 2nd highest energy atom produces the best classification accuracy. Learning basis features from speech, and then decomposing speech using these features appears to be a validate Gabor's theory that speech can be represented as a summation of "quanta's of information".

References

[1] A. S. Bregman, Auditory Scene Analysis. Cambridge, MA: MIT Press, 1990.
[2] Zhao, X., Shao, Y., Wang, D. CASA-Based Robust Speaker Identification IEEE transactions on audio, speech, and language processing, vol. 20, no. 5, July 2012
[3] Lee, H., Largman, Y., Pham, P., Ng, A. Unsupervised feature learning for audio classification using convolutional deep belief networks. Conference proceedings: Advances in Neural Information Processing Systems 22, 2009.
[4] Gabor, D., Theory of communication, J. Inst. Elect. Eng., 93, pp. 429–457. 1946.
[5] Mallat, S., Zhang, Z. Matching Pursuits with Time-Frequency Dictionaries. IEEE transactions on signal processing. Vol 41. No 12. 1993.
[6] Olshausen, B., Field, D., Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583):607–9, 1996.
[7] S. Haykin. Neural Networks and Learning Machines, third edition. Pearson Education, Inc. Prentice Hall 2009. Page 516.
[8] Grosse, R., Raina, R., Kwong, H., Ng, A., Shift-Invariant Sparse Coding for Audio Classification, UAI 2011.
[9] Bryan, T., Kepuska, V., Kostanic, I., A Simple Adaptive Atomic Decomposition Voice Activity Detector Implemented by Matching Pursuit, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:5, 2015.
[10] Bryan, T., Kepuska, V., Kostanic, I., Atomic Decomposition Audio Data Compression and Denoising using Sparse Dictionary Feature Learning, World Academy of Science, International Science Index vol:10 no:01