M-VECTORS: SUB-BAND BASED ENERGY MODULATION FEATURES FOR MULTI-STREAM AUTOMATIC SPEECH RECOGNITION

Samik Sadhu, Ruizhi Li, Hynek Hermansky

Center for Language and Speech Processing,
Johns Hopkins University,
Baltimore, MD, USA
{samiksadhu,ruizhili,hynek}@jhu.edu

ABSTRACT

In this paper, we propose a novel method to capture energy modulations from different frequency bands in speech into frame-level feature vectors, called Modulation-vectors or M-vectors, for use in Automatic Speech Recognition (ASR) systems. We show that in different multi-stream setups, with parallel streams for M-vectors and the popular Mel-frequency Cepstral Coefficient (MFCC) features, we can realize a boost in word recognition performance of end-to-end systems by $\approx 5\%$, and that of a monophone and triphone HMM-GMM ASR system by $\approx 18\%$ and $\approx 16\%$ respectively over using the traditional MFCC features.

***Index Terms*—** Automatic Speech Recognition, Feature Extraction, Modulation Spectrum, Hilbert Envelope, Multi-stream Automatic Speech Recognition

1. INTRODUCTION

Log Mel Bank features or Mel Frequency Cepstral Coefficient (MFCC) have become the standard features for Automatic Speech Recognition (ASR). However, these traditional features are frame based, each feature vector describing short (10-20 ms) segments of the signal. Typically, in an ASR, dynamic features representing feature differentials in the frame location (delta features) are appended to the static feature vector and several (currently up to about twenty or even more) adjacent frames of such static and dynamic features are typically concatenated together to describe the local spectral dynamics, in effect representing a form of modulation spectrum of speech. Although there has been a few prior works on the usefulness of modulation spectral features for speech recognition [1, 2, 3], its full potential in ASR is yet to be explored.

In this paper, we use a novel approach to compute the rate of change of energy in different sub-bands of speech (similar to the idea of modulation spectrum of speech) using Frequency Domain Linear Prediction (FDLP). We use the proposed M-vectors in different multi-stream settings to study its effect on speech recognition accuracy. The paper is organized as follows: In section 2, we discus in details, the theory behind our method of deriving the modulation spectrum and then in section 3 we go on to describe how we compute the frame-wise features using this technique. We describe our experimental setup in section 4 and illustrate our results in section 7 before we conclude our paper in section 6 with the important takeaways.

2. COMPUTING ENERGY MODULATIONS OF SPEECH

We derive the energy modulations of speech using autoregressive (AR) modeling of the squared magnitude of the analytic signal [4] (called *Hilbert Envelope*) of speech in K individual frequency sub-bands weighted by overlapping triangular windows in the mel frequency scale. The Hilbert Envelope (HE) represents the energy of the signal as a function of time and hence its AR approximation captures coarse variations in energy in each sub-band as a function of time. In this paper, we use approximations to this energy trajectory to derive speech features for use in automatic recognition of speech. Unlike traditional feature computation methods, we design the feature vector to captures the energy modulations over a much wider temporal context of 0.5 seconds. Thus, the proposed features have a neat way of modelling the temporal dynamics compared to simple concatenation of frame-wise features that is usually done at the front-end of ASR systems. To approximate the Hilbert Envelope of the signal in the individual frequency bands, we use the technique of Frequency Domain Linear Prediction (FDLP). [5]

2.1. Linear Prediction in time domain

The technique of Linear Prediction (LP) [6] has long been used in multiple fields of signal processing as a method of approximating the power spectrum of a discrete signal $x[n]$, n representing the time index, by the power spectrum of an autoregressive model. The degree of detail of the spectrum is controlled by choice of the all-pole model order p. This type of LP models are also termed Time Domain Linear Prediction

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

(TDLP).

2.2. Linear Prediction in frequency domain

The frequency domain equivalent of TDLP is FDLP, which was developed by Athineos [7] by applying the autocorrelation LP method on the Type-I Discrete Cosine Transform (DCT) $X[k]$ of the signal $x[n]$, k representing the frequency index as is typically denoted for DCT. Linear prediction of the sequence $X[k]$ results in an AR model of the Hilbert Envelope of the even-symmetrized version of the signal $x[n]$. A proof of this result can be found in [8]

The FDLP envelope provides a way of looking at the energy variations of a signal as a function of time with desired levels of approximation determined by the AR model order p. The filter gain G is computed as the squared error sum of the LP fit.

2.3. Approximating Hilbert Envelopes in individual frequency bands

FDLP can be also used to approximate Hilbert envelopes of parts of the signal spectrum. As proposed in [7], since the cosine transform of the signal $X[k]$ moves the problem into frequency domain, appropriate window on the $X[k]$ selects the part of the signal spectrum to be approximated.

2.4. Computing Energy Modulations by recursive cepstrum computation

We compute the energy modulation coefficients as the Discrete Fourier Transform (DFT) of the logarithm of the filter response to the inverse FDLP filter. It should be noted that this definition of the modulation coefficient makes it possible to compute them by using the recursive LP cepstrum computation formula [9]. This results in a significant improvement in computational efficiency for the features and the computation time is mostly dominated by computation of FDLP coefficients.

3. FRAME-WISE M-VECTOR COMPUTATION

In this section, we show the steps we follow to obtain frame-wise M-vectors for the front-end of an ASR system. Since typical Hidden Markov Model (HMM) based ASR systems require feature vectors about every 10 ms, we divide the signal into long windows of length 0.5 seconds and a frame-rate of 100 Hz. This ensures that we can obtain the M-vectors over a wide acoustic context and at the same time generate frame-level features at a rate expected by most ASR systems. We use 0.5 second Hanning window which decays down to zero at the edge of the window. This ensures that the M-vectors are computed with maximum emphasis over the modulations at the central quarter of each window. We use K triangular filters in the mel frequency scale to window the the DCT of the windowed signal into different sub-bands and obtain the energy modulation coefficients for each band. The final M-vector is obtained by concatenating the energy modulation coefficients from all the sub-bands.

3.1. Dealing with context at the edges

With such large windows, one challenge is to deal with data at the edges of the processed speech utterance. In general, since a considerably large portion around the edges of a speech utterance is silence, the problem is addressed by mirroring the silence segments at the edges of a speech file to provide for enough initial and final extension to the utterance to fit the long processing window.

3.2. M-vector parameters

M-vector computation has 4 important parameters to configure

- The number of modulation coefficients decides what range of rate of variation of energy is desired. Previous experiments have shown that the modulations in human speech are the most important in the range 2-8 Hz [10] with maximum linguistic information around 4 Hz. In our experiments, we observed that even the DC component of modulation has some information that is useful for speech recognition. Thus, we preserve the first 15 coefficients of energy modulations for our feature vectors to capture modulations in the range 0-8 Hz.

- The window size determines the temporal context over which the energy modulations are going to be computed. A high window length of 1 seconds or a small window length of 0.25 seconds appears to degrade the recognition performance and also captures modulations over different frequency ranges which are not relevant to acoustic modulations in speech.

- The FDLP order p determines the degree of approximation of the Hilbert Envelope. We use 100 poles per second for our experiments. We observed that the recognition performance did not change drastically as long as poles per second is not pushed to extremes.

- Number of sub-bands K appears to be no significant difference between recognition performance for 7,15 and 30 sub-bands. Hence, for computational efficiency we use 7 sub-bands for our experiments.

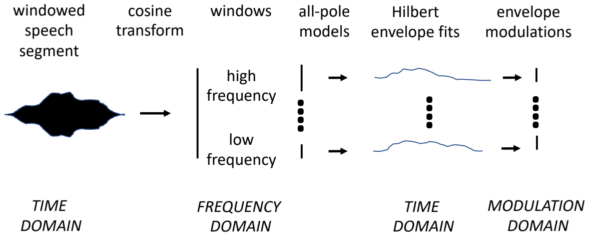

Fig. 1. Process of deriving M-vectors. A Hanning-windowed segment of speech is transformed through cosine transform into frequency domain. Properly shaped windows on the cosine transform of the signal emulate frequency filtering. The windowed segments are used for deriving all-pole models, which approximate Hilbert envelopes in the individual frequency bands. Concatenated rates of change (modulations) of the band-specific Hilbert envelopes form the feature vector describing the windowed speech segment.

3.3. Baseline features

We use standard 13 dimensional MFCC features for comparison with 15 triangular filters in the mel-scale for comparison with our M-vectors. It should be noted that, although we use the Kaldi ASR system [11] for all our experiments, we use our own feature computation codes to make sure that the features in comparison are perfectly time aligned and have the same number of frames.

4. EXPERIMENTAL SETUP

We validate the importance of the acoustic information in M-vectors by conducting experiments with HMM-GMM and end-to-end ASR systems.

4.1. HMM-GMM Systems

Using M-vectors directly in the standard HMM-GMM systems results in the following issues

1. MFCC features and M-vectors have different numerical ranges and hence it becomes difficult to avoid the ASR system to not get biased towards one of them when we simply concatenate MFCC features and M-vectors. Hence, it is necessary to bring the information from the two features to a similar numerical range.

2. The M-vectors are not normally distributed. Hence it is not very well modelled by HMM-GMM systems with diagonal covariance matrices.

4.1.1. Data-driven multi-stream feature generation

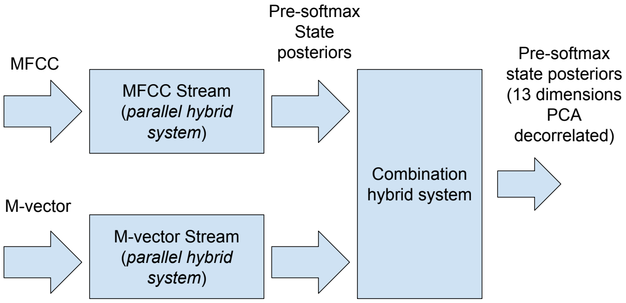
MFCC
M-vector
MFCC Stream (*parallel hybrid system*)
M-vector Stream (*parallel hybrid system*)
Pre-softmax State posteriors
Combination hybrid system
Pre-softmax state posteriors (13 dimensions PCA decorrelated)

Fig. 2. Combining the MFCC and M-vector feature streams

Multi-stream ASRs have gained much importance in speech recognition research in recent days [12, 13, 14, 15]. We use a multi-stream ASR system for combination of the two different feature streams into a single feature vector to be used in speech recognition. The multi-stream system is set up in the following way

- **Stage1:** We train a monophone HMM-GMM ASR system with MFCC features to generate phonetic alignments on the training data. We use the phonetic alignments to train individual *parallel* hybrid systems with MFCC and M-vectors. For MFCC we use ±4 context window, and for M-vectors we do not use any feature splicing. However, both features were mean and variance normalized before any other operations.

- **Stage2:** The presoftmax state posteriors from the DNN from the two streams above are concatenated into one feature vector which is used to train another *combination* hybrid system

- **M-vector-MFCC feature generation:** The presoftmax state posteriors from the *combination* DNN are reduced in dimension using Principal Component Analysis (PCA) to 13 dimensions and used as the final data-driven feature vector for ASR systems. We do not further splice these features for use in ASR systems because they have been derived over a large temporal context. We perform mean and variance normalization on these features before using them for ASR systems.

All the hybrid systems are trained with 5 layered DNNs with 256 nodes and tanh non-linearities using the Kaldi nnet2 recipe.

4.1.2. ASR details

We used Monophone and Triphone ASR systems with three state left-to-right HMMs for each phoneme from a basic set of 43 phonemes. The phonemes are split based on context and pronunciation emphasis into a set of 351 phonemes in the standard Kaldi recipe. We use a trigram language model alongside the acoustic model. All the ASR systems are trained using the Kaldi ASR system.

4.2. End-to-end multi-stream system

We set up an joint CTC-attention end-to-end system as proposed in [16], but with two encoders from two different feature streams - MFCC and M-vectors. The two encoders and CTC networks are of type BLSTM while the single decoder network is an LSTM. The streams are ultimately combined using a hierarchical attention mechanism and coupled to the decoder [17]. The final word error rate using the end-to-end system is obtained by decoding with a Recurrent Neural Network (RNN) language model.

5. RESULTS

We look at how the sub-band speech modulations in M-vector can be used alongside the usual MFCC features to boost recognition performance.

5.1. Database

All our experiments have been done using the entire Wall Street Journal(WSJ) database. Our ASR training and test sets consists of all the 37416 utterances ($\approx$ 82 hours) in `si284` and the 333 utterances in `eval92` respectively.

5.2. Usefulness of M-vectors

We use the M-vector-MFCC features from section 4.1.1 to train a monophone HMM-GMM system (section 4.1.2).

In order to verify that the M-vectors are indeed information bearing components which can be used to boost in the ASR performance, we use the presoftmax state posteriors from the DNN in the MFCC stream at **Stage 1** (dimension reduced and de-correlated by PCA) effectively generating Tandem features [18] from MFCC for comparison with the final combined features. Similar features can be obtained from the presoftmax state posteriors from the DNN in the M-vector stream. The results are shown in table 1.

Table 1. Usefulness of M-vectors with data-driven multi-stream features for HMM-GMM ASR systems

Feature Type	WER (%)
MFCC	26.28
MFCC Tandem	22.59
M-vector Tandem	27.40
M-vector-MFCC	21.64

The results from the multi-stream end-to-end system (section 4.2) is also shown in table 2.

Table 2. Usefulness of M-vectors in a multistream setup with MFCC for end-to-end ASR systems

E2E System	# Params	WER (%)
Single-Stream		
MFCC	15.2M	6.4
M-vector	15.4M	9.8
Multi-Stream		
MFCC+M-vector	14.6M	6.1

In end-to-end systems, all the model parameters are trained together. Hence, it is important to have the number of parameters of different models in the same range for effective comparison. Since the multi-stream system has two encoders, we use 8 encoder layers for the single-stream system and 4 encoder layers per stream for the multi-stream system to keep the number of parameters approximately same.

5.3. HMM-GMM ASR Performance

Table 3 compares the performance of Monophone and triphone ASR systems on the WSJ database for MFCC and M-vector-MFCC features (as described in section 4.1.1).

Table 3. Comparison of HMM-GMM ASR performance for MFCC and M-vector-MFCC features

ASR Systems	WER (%)	
	MFCC	M-vector-MFCC
Monophone	26.28	21.64
Triphone	13.22	11.11

.

6. CONCLUSIONS

The results from table 1 show that there is $\approx 4.2\%$ improvement in recognition accuracy over data driven MFCC-tandem features by merging the two feature streams. A similar improvement of $\approx 5\%$ is observed for a multi-stream end-to-end system processing the information from both feature streams together over a single stream end-to-end system with MFCC feature.

7. ACKNOWLEDGEMENT

The work was supported by the National Science Foundation under EAGER Grant No. 1743616, by a gift from Google Inc., and by JHU Human Language Technology Center of Excellence.

8. REFERENCES

[1] Brian ED Kingsbury, Nelson Morgan, and Steven Greenberg, “Robust speech recognition using the modulation spectrogram,” *Speech communication*, vol. 25, no. 1-3, pp. 117–132, 1998.

[2] Sriram Ganapathy, Samuel Thomas, and Hynek Hermansky, “Static and dynamic modulation spectrum for speech recognition,” in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[3] Sriram Ganapathy, Samuel Thomas, and Hynek Hermansky, “Modulation frequency features for phoneme recognition in noisy speech,” *The Journal of the Acoustical Society of America*, vol. 125, no. 1, pp. EL8–EL12, 2009.

[4] Lawrence Marple, “Computing the discrete-time” analytic” signal via fft,” *IEEE Transactions on signal processing*, vol. 47, no. 9, pp. 2600–2603, 1999.

[5] Marios Athineos and Daniel PW Ellis, “Autoregressive modeling of temporal envelopes,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5237–5245, 2007.

[6] John Makhoul, “Linear prediction: A tutorial review,” *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[7] Marios Athineos and Daniel PW Ellis, “Frequency-domain linear prediction for temporal features,” in *Automatic Speech Recognition and Understanding, 2003. ASRU’03. 2003 IEEE Workshop on*. IEEE, 2003, pp. 261–266.

[8] Sriram Ganapathy, *Signal analysis using autoregressive models of amplitude modulation*, Ph.D. thesis, Johns Hopkins University, 2012.

[9] Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai, “Recursive calculation of mel-cepstrum from lp coefficients,” *Trans. IEICE*, vol. 71, pp. 128–131, 1994.

[10] Hynek Hermansky, “The modulation spectrum in the automatic recognition of speech,” in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 140–147.

[11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[12] Sri Harish Mallidi and Hynek Hermansky, “Novel neural network based fusion for multistream asr,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5680–5684.

[13] Sri Harish Reddy Mallidi and Hynek Hermansky, “A framework for practical multistream asr.,” in *INTERSPEECH*, 2016, pp. 3474–3478.

[14] Sri Harish Mallidi, Tetsuji Ogawa, Karel Veselỳ, Phani S Nidadavolu, and Hynek Hermansky, “Autoencoder based multi-stream combination for noise robust speech recognition,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] Xiaofei Wang, Ruizhi Li, and Hynek Hermansky, “Stream attention for distributed multi-microphone speech recognition,” *Proc. Interspeech 2018*, pp. 3033–3037, 2018.

[16] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.

[17] Ruizhi Li, Xiaofei Wang, Sri Harish Mallidi, Takaaki Hori, Shinji Watanabe, and Hynek Hermansky, “Multi-encoder multi-resolution framework for end-to-end speech recognition,” *arXiv preprint arXiv:1811.04897*, 2018.

[18] Hynek Hermansky, Daniel PW Ellis, and Sangita Sharma, “Tandem connectionist feature extraction for conventional hmm systems,” in *icassp*. IEEE, 2000, pp. 1635–1638.